generating a native program for an application utilizing a set of native instructions having a first fixed number of bits;

debugging the native program;

processing the debugged native program by analyzing the set of native instructions at a sub-instruction level to determine specific patterns of bits that do not change within groups of instructions and utilizing the results of said analysis to determine an abbreviated instruction set having a second fixed number of bits less than the first fixed number of bits and corresponding to the set of native instructions; and

converting the native program to the abbreviated application specific program by replacing the set of native instructions with the abbreviated instruction set.

18. (amended) A method for generating an abbreviated instruction set corresponding to a set of native manifold array (ManArray) instructions for an application specific program comprising the steps of:

separating the set of native ManArray instructions into groups of instructions;

identifying the unique instructions within each group of instructions;

analyzing the unique instructions for common instruction characteristics;

determining at least one style pattern of bits which is defined as a specific pattern of bits that are constant; and

generating the abbreviated instruction set utilizing the at least one style.

26. (twice amended) A method for translating abbreviated instructions into a native instruction format comprising the steps of:

fetching an abbreviated instruction having a first fixed number of bits from a memory tailored to storage of abbreviated instructions;

dynamically translating the abbreviated instruction into the format of a native instruction by using a first bit field in the abbreviated instruction as an address reference to a first translation memory containing at least one specific sub-native instruction pattern of bits;

fetching a sub-native instruction pattern from the translation memory using said address reference, said sub-native instruction pattern being based on a previous analysis of the set of native instructions on a sub-instruction level to determine patterns of bits that do not change within groups of instructions;

combining the sub-native instruction patterns with bits from the abbreviated instruction to create the native instruction in a sequence processor (SP) array controller said native instruction having a second fixed number of bits greater than said first fixed number; and

dispatching the native instruction to the sequence processor array controller or a processing element for execution.

34. (twice amended) A system for translating abbreviated instructions into a native instruction format comprising:

a memory storing an abbreviated instruction having a first fixed number of bits;

means for fetching the abbreviated instruction from the memory;

means for dynamically translating the abbreviated instruction into a native instruction using a translation memory storing at least one specific sub-native instruction pattern of bits, said sub-native instruction pattern being based on a previous analysis of the set of native instructions on a sub-instruction level to determine patterns of bits that do not change within groups of instructions;

an addressing mechanism using a first bit field in the abbreviated instruction as an address reference to the translation memory;

means for fetching the sub-native instruction pattern from the translation memory; and

means for combining the sub-native instruction pattern with bits from the abbreviated instruction to create the native instruction in the native instruction format having a second fixed number of bits greater than said first fixed number.

Please add the following new claims:

57. A method for translating abbreviated instructions into a native instruction format comprising the steps of:

fetching an abbreviated instruction having a first fixed number of bits from a memory tailored to storage of abbreviated instructions;

dynamically translating the abbreviated instruction into the format of a native instruction by using a first and a second bit field in the abbreviated instruction as address references to a first field and a second translation memory each containing at least one specific sub-native instruction patterns of bits;

fetching a sub-native instruction pattern from each translation memory using said address references, each said sub-native instruction pattern being based on a previous analysis of the set of native instructions on a sub-instruction level to determine patterns of bits that do not change within groups of instructions;

combining the two sub-native instruction patterns to create the native instruction in a sequence processor (SP) array controller said native instruction having a second fixed number of bits greater than said first fixed number; and

dispatching the native instruction to the sequence processor array controller or a processing element for execution.

58. A system for translating abbreviated instructions into a native instruction format comprising:

a memory storing an abbreviated instruction having a first fixed number of bits;

means for fetching the abbreviated instruction from the memory;

means for dynamically translating the abbreviated instruction into a native instruction using two translation memories each storing at least one specific sub-native instruction patterns of bits, each of said sub-native instruction patterns being based on a previous analysis of the set of native instructions on a sub-instruction level to determine patterns of bits that do not change within groups of instructions;

two addressing mechanisms each using a bit field in the abbreviated instruction as an address reference to one of the two translation memories;

means for fetching the sub-native instruction patters from each translation memory; and

means for combining the sub-native instruction patterns to crate the native instruction in the native instruction format having a second fixed number of bits greater than said first fixed number.